INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A matrix-free preconditioner applied to CFD

Rémi Choquet

N° 2605

Juin 1995

PROGRAMME 6

ISSN 0249-6399

A matrix-free preconditioner applied to CFD

Rémi Choquet*

Programme 6 — Calcul scientifique, modélisation et logiciel numérique
Projet ALADIN

Rapport de recherche n° 2605 — Juin 1995 — 31 pages

Abstract: We consider the implicit time integration of the compressible Navier-Stokes equations. Each implicit step involves the solution of a costly non linear system. In this paper, we define a preconditioner to solve each linear system more cheaply within the framework of matrix-free algorithm. The main idea is to reuse the Krylov subspace across the time steps. We demonstrate some properties of the preconditioner and show the practical value of this approach for stiff problems which needs excessive small time steps with an explicit method.

Key-words: compressible Navier-Stokes, R-K and BDF schemes, Newton, GMRES, matrix-free, preconditioner.

(Résumé : tsvp)

*choquet@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Un préconditionneur sans matrice appliqué à la CFD

Résumé : Nous considérons l'intégration implicite en temps des équations de Navier Stokes compressibles. Chaque pas de temps nécessite la solution couteuse d'un système non-linéaire. L'approche considérée est celle où le problème est itérativement linéarisé. Dans ce papier, nous définissons un préconditionneur pour résoudre chaque système linéaire de manière efficace dans le cadre d'algorithme sans matrice. L'idée est de réutiliser le sous-espace de Krylov à travers les pas de temps. Nous montrons quelques propriétés de ce préconditionneur et les exploitons en montrant l'efficacité de ce préconditionneur appliquée entre autre à la simulation d'un problème raide.

Mots-clé : Navier-Stokes compressible, schémas R-K et BDF, Newton, GMRES, sans matrice, préconditionneur.

1 Introduction

The main motivation to solve the compressible Navier-Stokes equations by means of an implicit method is the excessive small time steps limited by stability constraints for explicit methods. Thus, for structured meshes, work has been done both on the development of new implicit schemes [LS88],[Sen90] and on the use of new tools to solve large sparse non-symmetric linear systems[WYY85],[HTS93]. Also for unstructured meshes, new schemes[HFM86],[SF94] and techniques [JHF91],[Dut91],[LF94],[Lan94] have been developped. However, in most cases, each implicit step leads to one costly non linear system to be solved.
When three-dimensional space problems and/or chemical context are considered, the size of the systems is usually large. Thus, the memory-size of current computers prohibits the storage of the matrix coming from the linearized problem. We are interested in finding a preconditioner for solving these linear systems when matrix-free algorithms are used. This preconditioner should be easy to compute both taking into account the miss of the matrix and the computer system. In particular, with a convenient data structure , the product of this preconditioner by a vector should be done easily in parallel. The problem cited previously occurs also in leading areas of numerical simulation such as Ocean and Climate modeling.

Let us describe more precisely the framework of the study where the following considerations take place. We have to solve

$$u_{,t} + G(u) = 0, \tag{1}$$

where $G(u) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a non-linear function of u. We consider either the stationary or non-stationary resolution of (1). Replacing $u_{,t}$ by a multi-step scheme, equation (1) leads at time t_n to

$$u(n+1) + \Delta t_n H_1(u(n+1)) = \Delta t_n H_2(u(n), u(n-1), u(n-2), \ldots). \tag{2}$$

We note that the backward Euler method corresponds to $H_2(u) = \frac{u}{\Delta t_n}$.
We refer to [CM89],[HW91] for a more complete description of ODE schemes. We suppose that H_1 is at least differentiable.
With $F(u) = u + \Delta t_n H_1(u) - \Delta t_n H_2(u(n), u(n-1), u(n-2), \ldots)$, (2) becomes

$$F(u(n+1)) = 0 \tag{3}$$

We choose to solve the nonlinear equation (3) by Newton-GMRES [BS90], where Newton iterations are applied to linearize the problem and the induced linear system is solved by the iterative GMRES scheme which only requires the Jacobian applied to a vector and is well-suited to non symmetric large matrices. The algorithm can be summarized as follow with $J = \frac{\partial F(u)}{\partial u}$.

ALGORITHM 1: Newton-GMRES

u_0 given, $i = -1$
REPEAT
  $i = i + 1$
  Solve $J(u_i)\delta u_i = -F(u_i)$ by GMRES
  $u_{i+1} = u_i + \delta u_i$
UNTIL convergence

Within GMRES, each Jacobian vector product is approximated by either a first or a second order finite difference scheme. Difficulties arising from such an approach include the following,

1. Finite difference schemes introduce errors which deteriorate both GMRES and Newton convergence. We refer to [CE93] for a study of the finite difference scheme GMRES-algorithms.

2. Restarting is one way to reduce the storage cost, but this approach decreases the convergence rate.

3. Another way is to find a preconditioner but traditional approaches like an incomplete LU factorisation are memory consuming.

We note that we are in the context where successive linear systems noted

$$J_i^{(n)} \delta u_i^{(n)} = -F_i^{(n)}$$

have to be solved by GMRES. With $J_i^{(n)}$ constant and symmetric, Saad[Saa87] study the reuse of the Krylov subspace to accelerate the convergence of the Lanczos process. More recently, Vuik[Vui93] used the same idea in the context of the solution of the incompressible Navier-Stokes equations by GMRESR[VdVV93]. He showed a good acceleration of the computation of the pressure by the linear solver. In [Cho94], we proposed a new algorithm to accelerate the Newton-GMRES algorithm. The main idea was to approximate the next Newton step from the Krylov base and the Hessenberg matrix computed by the Arnoldi process. An acceleration of the overall process was observed. In a recent paper [EBP95], a preconditioner was built to improve the restarted-GMRES algorithm. The preconditioner was based on an estimation of the small eigenvalue and its associated eigenspace. In this paper, with a similar approach, we build a cheap preconditioner from the Krylov base and the Hessenberg matrix computed by GMRES.

This paper is composed as follows. First in section 2, a short recall of the GMRES algorithm is given which outlines some notations. Then, we describe the preconditioner in section 3 and its theoretical contribution. Finally, in section 4, numerical results show the practical value of this approach in the context of CFD.

Notations : We denote by P_0^n the set of polynomials of degree up to n having the value 1 at 0, and by I_k the identity matrix of rank k.

2 GMRES

In this section, we present briefly the GMRES algorithm, introducing some notations we will need later.

Considering the system

$$J\delta = -F, \tag{4}$$

where $J = J(u)$, $F = F(u)$ and J is non-singular, the principle of GMRES is to minimize $||-J(\delta_0 + z) - F||_2$ with z in the Krylov subspace $K(J, r_0, k)$ defined next.

Definition 2.1 *The Krylov subspace $K(J, r_0, k)$ associated to the vector $r_0 = -F - J\delta_0$ and the matrix J is the subspace spanned by* $\mathrm{Span}\{r_0, Jr_0, \ldots, J^{k-1}r_0\}$ *where δ_0 denotes a first estimation of δ.*

Thus, (4) is replaced by the following minimization problem

$$\min_{z \in K(J, r_0, k)} ||r_0 - Jz||_2. \tag{5}$$

The Krylov subspace is generated by the Arnoldi process leading to the following GMRES algorithm.

ALGORITHM 2: GMRES (m)

given $\xi > 0$ the tolerance of the stopping criterion and $\delta_0 \in \mathbb{R}^N$ the initial guess,

(s1). $r_0 = -F - J\delta_0$,
$\beta = ||r_0||_2$, $v_1 = \dfrac{r_0}{\beta}$, $k = 0$

(s2). **repeat**
$k = k + 1$
(a) $w_{k+1} = Jv_k - \displaystyle\sum_{l=1}^{k} h_{l,k} v_l$
with $h_{l,k} = (Jv_k, v_l)$ $\quad (l = 1, \ldots, k)$
$h_{k+1,k} = ||w_{k+1}||_2$
$v_{k+1} = \dfrac{w_{k+1}}{h_{k+1,k}}$
(b) compute $\rho_k = \displaystyle\min_{y \in \mathbb{R}^k} ||\beta e_1 - \overline{H}_k y||_2$
until $\rho_k \leq \xi ||F||_2$ **or** $k = m$

(s3). compute y_k, $z_k = V_k y_k$ and $\delta_k = \delta_0 + z_k$

if $\rho_k > \xi ||F||_2$ **then**
  $\delta_0 = \delta_k$
  GOTO **(s1)**
end if

$$\text{where} \quad (\overline{H}_k)_{p,l} = \begin{cases} h_{p,l} & 1 \le p \le k+1, p \le l \le k \\ 0 & \text{otherwise} \end{cases}$$
$$V_k = (v_1, \ldots, v_k),$$

and

$$||\beta e_1 - \overline{H}_k y_k||_2 = \min_{y \in \mathbb{R}^k} ||\beta e_1 - \overline{H}_k y||_2.$$

The Gram-Schmidt process generates an orthogonal basis such that

$$JV_k = V_{k+1}\overline{H}_k, \tag{6}$$

implying

$$\min_{z \in K(J, r_0, k)} ||r_0 - Jz||_2 = \min_{y \in \mathbb{R}^k} ||\beta e_1 - \overline{H}_k y||_2. \tag{7}$$

3 The preconditioner

In our context, we have to solve sucessive linear systems

$$J_i^{(n)} \delta_i^{(n)} = -F_i^{(n)}.$$

Roughly speaking, our acceleration algorithm are based on the following idea : Compute at one step (n^* and i^* given) a preconditioner using $\overline{H}_k \in \mathbb{R}^{(k+1)\times k}$ and $V_{k+1} \in \mathbb{R}^{N\times(k+1)}$ satisfying (6) (with $J_{i^*}^{(n^*)} = J$) and to reuse it for $n \ge n^*$ and $i \ge i^*$.

In section 3.1, we first outline a basic acceleration algorithm. Then from the GMRES algorithm presented in the previous section, we build a preconditioner in subsection 3.2. In subsection 3.3, we study the convergence of GMRES applied to the preconditioned matrix. And a short variation taking into account the evolution of the Jacobian will be given in subsection 3.6 using a rank-one update.

3.1 Acceleration algorithm

We first give a basic algorithm of acceleration which aims at computing the preconditioner at one step and to reuse it for solving several linear systems.

ALGORITHM 3: Acceleration

$n = 0, \ldots$
  $u_0 = u(n)$
  $n^* = s(n)$
  $i = 0, \ldots$
    $i^* = r(i)$
    if op(i^*, n^*, i, n)=.true. **then**
      Solve $J_i^{(n)} G_{i^*}^{n^*} (G_{i^*}^{n^*})^{-1} \delta u_i = -F_i^{(n)}$
    else
      Solve $J_i^{(n)} \delta u_i = -F_i^{(n)}$
    end if
    if oq(i^*, n^*, i, n) **then** build G_i^n
    $u_{i+1} = u_i + \delta u_i$
  $u(n+1) = u_{i+1}$

3.2 Definition of the preconditioner

We consider the linear system $J\delta = -F$ and assume that we applied k iterations of GMRES starting with $r_0 = -F - J\delta_0$, so that we get a Krylov basis V_{k+1} and a Hessenberg matrix $\overline{H}_k$ such that $JV_k = V_{k+1}\overline{H}_k$ with the following QR factorisation

$$\begin{aligned} \overline{H}_k &= Q_k R_k, \\ &= \overline{Q}_k \overline{R}_k = (Q_k, \overline{q}_{k+1}) \begin{pmatrix} R_k \\ 0 \end{pmatrix}. \end{aligned}$$

Now, we define a preconditioner $C(J, F)$, noted C here for the matrix J. The objective is to solve J exactly on the subspace spanned by $V_{k+1}Q_k$.

Definition 3.1 *Let* $\alpha \in \mathbb{R}, \alpha \neq 0$,

$$C = V_{k+1} \left\{ \begin{pmatrix} \alpha R_k^{-1} & 0 \\ 0 & 1 \end{pmatrix} \overline{Q}_k^t - I_{k+1} \right\} V_{k+1}^t + I_N .$$

With this definition, we show that the preconditioned matrix JC has k eigenvalues equal to α associated to the invariant subspace spanned by $V_{k+1}Q_k$. More precisely, we have the following theorem:

Theorem 3.1 *Let the orthogonal base* P' *of* $\mathbb{R}^N$ *be*

$$\begin{aligned} P' &= (V_{k+1}\overline{Q}_k, W) \\ &= (V_{k+1}Q_k, V_{k+1}\overline{q}_{k+1}, W) \\ &= (V_{k+1}Q_k, W_2) \end{aligned}$$

then

$$JC = P' \begin{pmatrix} \alpha I_k & J_{12} \\ 0 & J_{22} \end{pmatrix} P'^t, \tag{8}$$

where $J_{12} = (V_{k+1}Q_k)^t JW_1$ *and* $J_{22} = W_2^t JW_1$ *with* $W_1 = (v_{k+1}, W)$.

Proof. (theorem) Let the orthogonal base P of $\mathbb{R}^N$ be

$$\begin{aligned} P &= (V_{k+1}, W), \\ &= (V_k, W_1). \end{aligned}$$

Rewriting J in these new basis P and P', lead to the following lemma.

Lemma 3.1 $P'^t JP = \begin{pmatrix} R_k & J_{12} \\ 0 & J_{22} \end{pmatrix}$.

Proof. (lemma) From (6), we have

$$(V_{k+1}Q_k)^t JV_k = R_k. \tag{9}$$

$$\begin{aligned} \text{Thus} \quad P'^t JP &= (V_{k+1}Q_k, W_2)^t J(V_k, W_1) \\ &= \begin{pmatrix} R_k & (V_{k+1}Q_k)^t JW_1 \\ W_2^t JV_k & W_2^t JW_1 \end{pmatrix} \\ &= \begin{pmatrix} R_k & J_{12} \\ 0 & J_{22} \end{pmatrix} \end{aligned}$$

□

Furthermore, C can be rewriten as

Lemma 3.2 $C = PBP'^t$ *with* $B = \begin{pmatrix} \alpha R_k^{-1} & 0 \\ 0 & I_{N-k} \end{pmatrix}$.

Proof. From the definition 3.1,

$$\begin{aligned} C &= V_{k+1} \left\{ \begin{pmatrix} \alpha R_k^{-1} & 0 \\ 0 & 1 \end{pmatrix} \overline{Q}_k^t - I_{k+1} \right\} V_{k+1}^t + I_N \\ &= V_{k+1} \begin{pmatrix} \alpha R_k^{-1} & 0 \\ 0 & 1 \end{pmatrix} (V_{k+1}\overline{Q}_k)^t - V_{k+1}V_{k+1}^t + I_N \\ &= V_{k+1} \begin{pmatrix} \alpha R_k^{-1} & 0 \\ 0 & 1 \end{pmatrix} (V_{k+1}\overline{Q}_k)^t + WW^t. \end{aligned}$$

□

From lemma 3.1, we have

$$P'^t JPB = \begin{pmatrix} \alpha I_k & J_{12} \\ 0 & J_{22} \end{pmatrix}$$

and the theorem follows. □

In the next subsection, we study the rate of convergence of GMRES applied to $A = JC = P'\widehat{A}P'^t$. We show how to choose α to optimize the preconditioner, in relation to $W_2^t JW_1$.

3.3 Properties

Assuming that we want to solve $Jx = R_0$ starting with an initial residual R_0, we study the residual R_l after l iterations of GMRES applied to $JC = A$. Let $\widehat{R}_0 = P'^t R_0$. By the orthogonality of P', we first have

$$\|\widehat{R}_0\|_2 = \|R_0\|_2 \tag{10}$$

The main result of this subsection is based on the following lemma.

Lemma 3.3 *If $(\alpha I_{N-k} - J_{22})$ is non singular then $\exists X \in \mathbb{R}^{N\times N}$ non singular such that*

$$X^{-1}\widehat{A}X = \begin{pmatrix} \alpha I_k & 0 \\ 0 & J_{22} \end{pmatrix} = \widetilde{A}$$

Proof. We take $X = \begin{pmatrix} I_k & X_{12} \\ 0 & I_{N-k} \end{pmatrix}$ with $X_{12} = -J_{12}(\alpha I_{N-k} - J_{22})^{-1}$. Then, we can easily check that

$$X^{-1}\widehat{A}X = \begin{pmatrix} \alpha I & 0 \\ 0 & J_{22} \end{pmatrix}$$

□

Now, we will measure the speed-up of GMRES that we can obtain with an appropriate use of the preconditioner. First, we need some additional notations. In the X base, $\widehat{R}_0 = X\gamma$ where $\gamma = (\gamma_1, \gamma_2)^t$ is partitioned correspondingly to the partition of $X = (X_1, X_2)$. Let $\widehat{R}'_0 = X\gamma'$ where $\gamma' = (0, \gamma'_2)^t$ with γ'_2 later defined. We consider two GMRES processes applied to $\widehat{A}$ starting respectively from $\widehat{R}_0$ and $\widehat{R}'_0$. We note by extension $\widehat{R}_l$ and $\widehat{R}'_l$ the residual after m iterations of GMRES. Then, we get the proposition.

Proposition 3.1 *If $(\alpha I_{N-k} - J_{22})$ is non singular then*

$$\begin{aligned} \|\widehat{R}_{l+1}\|_2 \leq \|\widehat{R}'_l\|_2 \leq & \quad \|p(\widehat{A})\widehat{R}'_0\|_2 \\ \leq & \quad \|Xp(\widetilde{A})\gamma'\|_2 \end{aligned}$$

for all $p \in P_0^l$ with $\gamma' = (0, (I_{N-k} - \alpha^{-1}J_{22})\gamma_2)^t$.

Proof. We have

$$\begin{aligned} \|\widehat{R}_{l+1}\|_2 &= \min_{p\in P_0^{l+1}} \|p(\widehat{A})\widehat{R}_0\|_2 \\ &\leq \|p(\widehat{A})h_1(\widehat{A})\widehat{R}_0\|_2 \text{ with } h_1 \in P_0^1 \\ &\leq \|p(\widehat{A})XX^{-1}h_1(\widehat{A})\widehat{R}_0\|_2 \end{aligned}$$

As

$$X^{-1}\widehat{A}\widehat{R}_0 = \alpha\gamma_1 + J_{22}\gamma_2,$$

we take $h_1(x) = 1 - \alpha^{-1}x$. We obtain

$$X^{-1}h_1(\widehat{A})\widehat{R}_0 = X^{-1}\widehat{R}_0 - \alpha^{-1}X^{-1}\widehat{A}\widehat{R}_0 = X^{-1}X_2\gamma'_2 = X^{-1}\widehat{R}'_0 = \gamma',$$

with $\gamma_2' = (I_{N-k} - \alpha^{-1} J_{22})\gamma_2$.
So

$$\begin{aligned} \|\widehat{R}_{l+1}\|_2 &\leq \|Xp(\widetilde{A})\gamma'\|_2 \\ &\leq \|p(\widehat{A})\widehat{R}_0'\|_2. \end{aligned}$$

In particular,

$$\|\widehat{R}_{l+1}\|_2 \leq \|\widehat{R}_l'\|_2.$$

which encloses the proof. □

We deduce easily from (10) a corollary of the proposition (3.1).

Corollary 3.1 *If $(\alpha I_{N-k} - J_{22})$ is non singular then $R_{l+1} = P'\widehat{R}_{l+1}$ satisfy $\forall\ p \in P_0^l$*

$$\|R_{l+1}\|_2 \leq \|\widehat{R}_l'\|_2 \leq \|X\gamma''\|_2$$

with $\gamma'' = (0, \gamma_2'')^t,\ \gamma_2'' = p(J_{22})\gamma_2'$.

Now, we are able to link to $W_2^t J W_1$ the convergence of the GMRES method applied to A.

Theorem 3.2 *Assuming that $(\alpha I_{N-k} - J_{22})$ is non singular then*

$$\|R_{l+1}\|_2 \leq (\|X_{12}\|_2^2 + 1)^{\frac{1}{2}} \min_{p \in P_0^l} \|p(J_{22})\gamma_2'\|_2$$

with $J_{12} = (V_{k+1}Q_k)^t J W_1,\ J_{22} = W_2^t J W_1,\ \gamma_2' = (I - \alpha^{-1}J_{22})\gamma_2,\ X_{12} = -J_{12}(\alpha I_{N-k} - J_{22})^{-1}$ and $\|R_{l+1}\|_2 = \min_{p \in P_0^{l+1}} \|p(A)P'X\gamma\|_2$.

Proof. We have,

$$\begin{aligned} \|X\gamma''\|_2^2 &= \|Xp(J)\gamma'\|_2^2 \\ &= \left\| \begin{pmatrix} X_{12}p(J_{22})\gamma_2' \\ p(J_{22})\gamma_2' \end{pmatrix} \right\|_2^2 \\ &\leq (\|X_{12}\|_2^2 + 1)\|p(J_{22})\gamma_2'\|_2^2 \end{aligned}$$

The theorem follows when we apply the Corollary 3.1. □

Now, we can discuss the theoretical results obtained and the choice of the free parameter α. First, we consider the rate of convergence of GMRES with the preconditioned matrix A. We choose α such that $\|J_{12}\|_2 \leq \frac{\alpha}{2}$ and $\|J_{22}\|_2 \leq \frac{\alpha}{2}$. Thus, $(\alpha I - J_{22})$ is non-singular and $\|X_{12}\|_2 \leq 1$. Furthermore, γ_2 is independant of α. Thus, in Theorem 3.2, for α large enough, we have shown that the convergence is related to $J_{22} = W_2^t J W_1$. But as k is in general small compare to N, the condition number of $J_{22} = W_2^t J W_1$ is not guarantee to be smaller than the condition number of J. Nevertheless, if R_0 is close to r_0 then we expect

$$\|\gamma_2'\| \leq \frac{3}{2}\|\gamma_2\| << \|\gamma\|.$$

In the next section, we will compare the GMRES Algorithm applied to J and A.

3.4 Comparison

Now, we study the GMRES algorithm applied to J for the same linear system $Jx = R_0$ with the same initial residual R_0. And we compare this process with GMRES applied to A under the following restrictive assumptions.

(A1)- $\text{Span}\{V_k\} = \text{Span}\{K(J, r_0, k)\}$,

(A2)- $\|R_0 - \frac{(r_0,R_0)r_0}{(r_0,r_0)}\|_2 \leq tol\|R_0\|_2$,

(A3)- $\|J^i \left(R_0 - \frac{(r_0,R_0)r_0)}{(r_0,r_0)}\right) \|_2 \leq \|J^{i-1} \left(R_0 - \frac{(r_0,R_0)r_0}{(r_0,r_0)}\right) \|_2$.

Let first begin with the initial algorithm. We denote by r_l the residual after l iterations of GMRES applied to J.

Proposition 3.2 *Under assumptions (A1) to (A3), we have*

$$\|r_{l+k}\|_2 \leq \min_{p \in P_0^l} |\theta_0| \|p(J)\tau_k(J)r_0\|_2 + tol(1 + \sum_{i=2}^{k} |\tau_{k,i}|)\|R_0\|_2$$

with $\theta_0 = \frac{(r_0,R_0)}{\|r_0\|_2^2}$ *and* $\tau_k(J)r_0 \perp span(V_{k+1}Q_k)$, $\tau_k(x) = 1 + \sum_{i=2}^{k} \tau_{k,i}x^i$ *solution of the problem*

$$\min_{p \in P_0^k} \|p(J)r_0\|_2.$$

Proof.

$$\begin{aligned}
\|r_{l+k}\|_2 &= \min_{p \in P_0^{l+k}} \|p(J)R_0\|_2 \\
&= \min_{p \in P_0^{l+k}} \|p(J)(\theta_0 r_0 + (I - \frac{r_0 r_0^t}{\|r_0\|_2^2})R_0)\|_2 \\
&\leq \min_{p \in P_0^{l}} |\theta_0| \|p(J)\tau_k(J)r_0\|_2 + \|\tau_k(J)(R_0 - \theta_0 r_0)\|_2 \\
&\leq \min_{p \in P_0^{l}} |\theta_0| \|p(J)\tau_k(J)r_0\|_2 + \sum_{i=1}^{k} \|\tau_{k,i}J^i(R_0 - \theta_0 r_0)\|_2 \\
&\leq \min_{p \in P_0^{l}} |\theta_0| \|p(J)\tau_k(J)r_0\|_2 + (1 + \sum_{i=2}^{k} |\tau_{k,i}|)\|R_0 - \theta_0 r_0\|_2
\end{aligned}$$

Now, we prove that $\tau_k(J)r_0 \perp \text{Span}\{V_{k+1}Q_k\}$. We suppose that it is not true, thus there exists $y \in \mathbb{R}^k$ such that $(\tau_k(J)r_0, V_{k+1}Q_k y) \neq 0$. Furthemore, there exits α such that

$$\|\tau_k(J)r_0 - \alpha V_{k+1}Q_k y\|_2 \leq \|\tau_k(J)r_0\|_2$$

which contradict the minimum property of τ_k. □

Now, we compare the GMRES processes applied to J and A under the assumptions $(A1)$ to $(A3)$. When α is large then $\gamma_2' \approx \gamma_2$ and $X_{12} \approx 0$ thus $\widehat{R}_l' \approx (0, I_{N-k} q_l(J_{22})\gamma_2)^t$ and becomes almost orthogonal to Span$\{V_{k+1}Q_k\}$. Thus, we can compare the results of proposition 3.2 and corollary 3.1. Both results give upper bound residuals which are almost orthogonal to Span$\{V_{k+1}Q_k\}$. The first one needs k iterates to be attained whereas the second one needs one iterate.
But, in our context, the Jacobian is varying both during the Newton and the time iterations. So, in the next subsection, we refer to a well-known result of perturbation to justify the small variation of the invariant subspace for a small perturbation of the Jacobian. Then in section 3.6, we propose an update of our preconditioner by the Broyden algorithm, i.e when the variation of J is a rank-one update.

3.5 Perturbation

Now, we will look at the evolution of the invariant subspace Span$\{V_{k+1}Q_k\}$ when A is perturbed. Let $E = \begin{pmatrix} E_1^t \\ E_2^t \end{pmatrix}$ be a perturbation of J the jacobian with $E_1 \in \mathbb{R}^{N\times k}$ and $E_2 \in \mathbb{R}^{N\times(N-k)}$. EC is the perturbation for the preconditioned matrix.

$$EC = \begin{pmatrix} E_1^t(\alpha V_k R_k^{-1}(V_{k+1}Q_k)^t + W_1 W_2^t) \\ E_2^t(\alpha V_k R_k^{-1}(V_{k+1}Q_k)^t + W_1 W_2^t) \end{pmatrix}$$

thus, $P'^t ECP' = \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}$ with $\begin{cases} E_{11} = \alpha(V_{k+1}Q_k)^t E V_k R_k^{-1} \\ E_{12} = (V_{k+1}Q_k)^t E W_1 \\ E_{21} = \alpha(W_2)^t E V_k R_k^{-1} \\ E_{22} = (W_2)^t E W_1 \end{cases}$

Proposition 3.3 *Let $\|.\|$ a norm and set*

$$\begin{aligned} a &= \|E_{21}\|, \\ b &= \|J_{12}\| + \|E_{12}\|, \\ c &= sep(\alpha I_k, J_{22}) - \|E_{11}\| - \|E_{22}\|. \end{aligned}$$

where $sep(\alpha I_k, J_{22}) = \inf_{\|P\|=1, P\in\mathbb{R}^{(N-k)\times k}} \|\alpha P - J_{22}P\|$.
If $c > 0$ and $\frac{ab}{c^2} < \frac{1}{4}$, there is a unique matrix P satisfying $\|P\| \le 2\frac{a}{c}$ such that

$$(V_{k+1}Q_k + W_2 P)(I + P^H P)^{-\frac{1}{2}}$$

form a orthonormal bases for the right invariant subspace of $(J+E)C$.

Proof. This proposition is a direct consequence of theorem 3.1 and theorem 2.7 p 236 of [SS90]. □

When α becomes high then the existence of P is guaranteed; furthermore $\|P\|$ is bounded independently from α. Thus, the perturbation of Span$\{V_{k+1}Q_k\}$, an invariant subspace of A, is small when $\|E\|$ is small enough.

3.6 Rank-one update of the preconditioner

During the Broyden algorithm [DS83], the Jacobian matrix is changing with a rank-one update. We propose an update of the preconditioner of the subsection 3.2 to take into account this evolution. We consider a rank one update of J, $J + xy^t$ and we want to define a similar preconditioner on $J+xy^t$. For the Broyden algorithm, $x = \frac{F(u_{i+1})-F(u_i)-J\delta u_i}{\|\delta u_i\|_2^2}$, $y = \delta u_i$.

We assume that $x \notin span(V_{k+1})$. We define $V' = (V_{k+1}, v')$ with $v' = \frac{(I_N - V_{k+1}V_{k+1}^t)y}{\|(I_N - V_{k+1}V_{k+1}^t)y\|_2}$.
We have

$$\begin{aligned}
(J + xy^t)V_k &= V'\left(\begin{pmatrix} \overline{H}_k \\ 0 \end{pmatrix} + \theta\tilde{y}^t\right) && \text{where } V'\theta = x \text{ and } \tilde{y} = V_k^t y \\
&= V'Q_kR_k && \text{where } Q_k \in \mathbb{R}^{(k+2)\times k} \text{ and } R_k \in \mathbb{R}^{k\times k} \\
&= V'Q_{k+2}\begin{pmatrix} R_k \\ 0 \\ 0 \end{pmatrix} && \text{where } Q_{k+2} = (Q_k, q_{k+1}, q_{k+2}) \in \mathbb{R}^{(k+2)\times(k+2)}
\end{aligned}$$

$$\Longrightarrow (V'Q_k)^t(J + xy^t)V_k = R_k \tag{11}$$

We define the new preconditioner C as

Definition 3.2

$$C = V'\left\{\begin{pmatrix} \alpha R_k^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} Q_{k+2}^t - I_{k+2}\right\} V'^t + I_N,$$

with $\alpha > 0$.

It satisfies the following theorem.

Theorem 3.3 *Let the orthogonal base* P' *of* $\mathbb{R}^N$ *be*

$$\begin{aligned}
P' &= (V'Q_{k+2}, W) \\
&= (V'Q_k, V'q_{k+1}, V'q_{k+2}, W) \\
&= (V'Q_k, W_2)
\end{aligned}$$

then

$$(J + xy^t)C = P'\begin{pmatrix} \alpha I_k & J_{12} \\ 0 & J_{22} \end{pmatrix} P'^t.$$

where $J_{12} = (V'Q_k)^t(J + xy^t)W_1$ *and* $J_{22} = W_2^t(J + xy^t)W_1$ *with* $W_1 = (v_{k+1}, v', W)$.

Proof. Let the orthogonal base P of $\mathbb{R}^N$ be

$$\begin{aligned}
P &= (V', W), \\
&= (V_k, W_1).
\end{aligned}$$

The proof is a direct consequence of the two following Lemmas. □

Lemma 3.4 *Let* $B = \begin{pmatrix} \alpha R_k^{-1} & 0 \\ 0 & I \end{pmatrix}$ *then* $C = PB'P'^t$.

Proof. From the definition 3.2,

$$\begin{aligned} C &= V' \left\{ \begin{pmatrix} \alpha R_k^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} Q_{k+2}^t - I_{k+2} \right\} V'^t + I_N \\ &= V' \begin{pmatrix} \alpha R_k^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} (V'Q_{k+2})^t - V'V'^t + I_N \\ &= V' \begin{pmatrix} \alpha R_k^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} (V'Q_{k+2})^t + WW^t. \end{aligned}$$

□

Lemma 3.5 $P'^t(J + xy^t)P = \begin{pmatrix} R_k & J_{12} \\ 0 & J_{22} \end{pmatrix}$

Proof. The proof is similar to the proof of lemma (3.1) once we have checked that $W_2^t(J + xy^t)V_k = 0$.

$$\begin{aligned} W_2^t(J + xy^t)V_k &= W_2^t V'Q_k R_k \\ &= \begin{pmatrix} (V'q_{k+1})^t V'Q_k R_k \\ (V'q_{k+2})^t V'Q_k R_k \\ W^t V'Q_k R_k \end{pmatrix} \\ &= 0 \end{aligned}$$

□

3.7 General remarks on the preconditioner

Choice of the parameter α. In the practical investigations, we will choose α as the largest singular value of the Hessenberg matrix $H_k = V_k^t J V_k$.

Cost of the preconditioner. In the framework of the iterative methods for solving the linear systems, the efficiency of the preconditioner C defined in subsection 3.1 depends also on the cost of the product of C by a vector v. Once the R_k and Q_k are built, we need $k + 1$ dot-products, $k + 2$ saxpy, $k + 2$ Givens rotations and one solution of a triangular system of size k to compute Cv. The overall cost is cheap compared to the computation of $F(u)$. We get almost the same cost for the preconditioner C defined in subsection 3.2. Now, we consider the cost of the preconditioner in terms of memory. As n is large compared to m, the cost is evaluated as the number of vectors that we need to store. Each time we build a preconditioner, we have to keep $k + 1$ vectors in memory.

A two level preconditioner. Let D be a preconditioner of the Jacobian, we assume that D is non-singular. If we consider a right preconditioning of J by D, we solve $(JD)y = F$ by GMRES. Instead of (6), we have

$$(JD)V_k = V_{k+1}\overline{H}_k.$$

As in section 3, we can build a preconditioner C for JD and DC is a preconditioner for J.

4 Practical investigations in CFD

We consider the implicit solution of a compressible, Newtonian and viscous fluid without source terms. The Navier-Stokes equations governing the flow are written in the following conservative form :

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho U) = 0 \\ &\frac{\partial (\rho U)}{\partial t} + \mathrm{div}(\rho U \otimes U) + \nabla p = \mathrm{div}(\mu S) \\ &\frac{\partial e}{\partial t} + \mathrm{div}((e+p)U) = \mathrm{div}(\mu S U) + \mathrm{div}(\kappa \nabla T) \end{aligned} \tag{12}$$

where ρ is the density; U the velocity ; T the temperature ; $e = \rho(T + \frac{\|U\|^2}{2})$ the total energy density; $p = (\gamma - 1)\rho T$ the pression and $S = (\nabla U + \nabla U^t) - \frac{2}{3} div(U) I$ the deformation tensor. $\kappa = \frac{\gamma \mu}{Pr}$ is defined from the following parameters $\gamma = 1.4$; $Pr = 0.72$ for the air; and $\mu = \frac{1}{Re}$ is given by the Sutherland law :

$$\mu = \mu_\infty \left(\frac{T}{T_\infty} \right)^{1.5} \left(\frac{T_\infty + 110}{T + 110} \right),$$

where the subscript ∞ denotes reference quantities.
To solve (12), we use the conservative variables $(\rho, \rho U, e)$ [Moh94]. The convective terms are upwinded thanks to a finite volume formulation for the Euler part of the equations. The Riemann problem at each interior interface is solved by approximating the flux with the Osher scheme [OC82] or the Roe scheme [Roe81]. This scheme is differentiable if and only if $U.\nu \neq 0$ (where U and ν are respectively the value of the velocity and the normal at each interface), so we use the C_1 Steger-Warning flux splitting [SW81] to approximate the flux at the free-stream boundary. For the diffusive term, we use a standart P_1 finite element formulation. Roughly speaking, after a mixed finite volume/finite element P_1 formulation, we have to solve (1), where u is composed of blocks of the variables $(\rho, \rho U, e)$.

In the framework of unstructured meshes, we will consider the flux approximation as a black box and we will apply an implicit method to solve (1) leading to the non linear problem (2). Since the Jacobian is not explicitly known, we use an inexact matrix-free algorithm. The

algorithm is Newton-GMRES where each matrix-vector product $J(u)v$ is approximated by a finite difference scheme [CE93] like

$$J(u)v \approx \frac{F(u+\sigma v) - F(u)}{\sigma}.$$

In this context, we apply the preconditioner defined previously for the GMRES method. In this section, we consider the practical value of the acceleration algorithm presented in section 3.2. First, we consider the local contribution of the preconditioner for two steady-state problems, we consider its efficency both across the Newton iterations and the time steps. We will show that the preconditioner is efficient across the time steps. Then, we consider a full computation of a subcritical flow where implicit methods are better than explicit methods. Finally, in part 4.2.1, we consider the computation of an unsteady problem. Some discussions will be done first on the parameters of the algorithm and a full computation of the stiff part of this problem will be presented. We will show that this preconditioner is always valid.

4.1 Local tests for two steady problems

We look in details the practical contribution of the preconditioner in order to settle the «best strategy »for the choice of (i^*, n^*). We consider only steady-state solutions of (1) across a few time steps with a basic time step integrator : A backward Euler integrator. We used the software developed at INRIA [Moh94] where time is discretised by an explicit Euler scheme, given by

$$u(n+1) = u(n) + \Delta t_n G(u).$$

Then, we have implemented an implicit backward Euler scheme, given by :

$$u(n+1) + \Delta t_n G(u(n+1)) - u(n) = 0, \tag{13}$$

where each time step gives rise to the nonlinear problem $F(u) = 0$ where $F(u) = u + \Delta t_n G(u) - u(n)$. Assuming that $\|\frac{\partial G(u)}{\partial u}\|$ is bounded, we can always find a time step Δt_n such that the Jacobian $J(u) = I + \Delta t_n \frac{\partial G(u)}{\partial u}$ of (13) is non singular. Since the Jacobian is not explicitly known, we solve (13) by the inexact matrix-free algorithms , including a backtracking strategy. For numerical purposes, we consider the following two steady-state problems in a two-dimensional space :

Problem 1. a viscous flow at Mach 0.8, Re=5000 around a Naca0012 without incidence.

Problem 2. an inviscid flow at Mach 1.2, around a Naca0012 with an incidence of seven degrees.

We discretise the domain by a finite element mesh with 801 nodes and 1516 triangles, thus $N = 3204$. All the computations are done on a SPARC-IPX workstation.
The explicit Euler scheme is used during the 100 first steps where the solution varies greatly, with a local time step $\Delta t_n(exp)$. Then reasonably large time steps $\Delta t_n(imp) =$

$CFL.\Delta t_n(exp)$ can be used in the implicit scheme while guaranteeing the convergence of the Newton iterations. The Euler flux is approximated here by the Osher scheme. We will measure the efficiency of the precondionner in two directions ;

1. across the Newton iterations in subsection 4.1.1 and
2. across the time steps in subsection 4.1.2.

Notes for the Tables in subsections 4.1.1 and 4.1.2
The tables are organized by columns where the contents of columns from 1 to 7 are respectively

- 1 - $i->$ the state of the Newton process or
$n->$ the time step,

- 2,5 - the non-linear relative residual,

- 3,6 - the linear residual of GMRES(m),

- 4,7 - the number of Jacobian vector products.

4.1.1 Across Newton iterations

We consider one time step and we measure the contribution of the preconditioner without update across the Newton step. At each stage i of Newton, we evaluate a preconditioner C_i for the matrix $J_i \mathrm{G}_{i-1}$ where

$$\mathrm{G}_{-1} = I \quad \text{and} \quad \mathrm{G}_i = \mathrm{G}_{\mathrm{i}-1} C_i, \ i \geq 0.$$

The new preconditioner G_i is a composition of the preconditioners C_i. Then algorithm (3) becomes

ALGORITHME 4: acc-new

$n = 0, \ldots$
  $u_0 = 0$; $G_{-1} = I$; $n^* = n$;
  $i = 0, \ldots$
    $i^* = i - 1$
    Solve $J_i^{(n)} G_{i^*} (G_{i^*})^{-1} \delta u_i = -F_i^{(n)}$ by GMRES(m)
    Compute $C_i(J_i^{(n)} G_{i^*}, F_i^{(n)})$
    $G_i = G_{i-1} C_i$
    $u_{i+1} = u_i + \delta u_i$
  $u(n+1) = u_{i+1}$

In tables 1 and 2, we evaluate the preconditioner at the first cycle of GMRES of each Newton stage. We see that a slight improvement has been achieved during the GMRES process which entails a better convergence of the Newton process. We also considered the preconditioner of subsection 3.6 with the Broyden update as a rank one update [DS83, Cho94]. But experimental results showed no improvement of the convergence compared to the original preconditioner.

				Acceleration		
i	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\|r\|_2}{\|F(u_0)\|_2}$	J*v	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\|r\|_2}{\|F(u_0)\|_2}$	J*v
0	1.0	5.469117E-05	6	1.0	5.469298E-05	6
1	6.312579E-05	2.116724E-04	10	6.312579E-05	1.927194E-04	10
2	1.336176E-08	2.444435E-04	10	1.216542E-08	2.043502E-04	10
3	5.460306E-12			5.12953E-12		

Table 1: Observations for problem 1 when $m = 10$ and $CFL = 1$.

				Acceleration		
i	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\|r\|_2}{\|F(u_0)\|_2}$	J*v	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\|r\|_2}{\|F(u_0)\|_2}$	J*v
0	1.0	8.688418E-04	11	1.0	8.688418E-04	11
1	1.350409E-02	7.699990E-04	23	1.350409E-02	8.904531E-04	20
2	1.048916E-05	9.204466E-04	22	1.216324E-05	6.755954E-04	22
3	3.792668E-08			8.315497E-09		

Table 2: Observations for problem 2 when $m = 15$ and $CFL = 5.0$.

4.1.2 Across the time steps

ALGORITHME 5: acc-time(opt)

given $freq$,
$n = 0, \ldots$
  $u_0 = 0$; $G_{-1}^{(n)} = I$; $n^* = n - \mathrm{mod}(n, freq)$;
  $i = 0, \ldots, i_{max} - 1$
    $i^* = i$,
    if $\mathrm{mod}(n, freq) = 0$ **then**
      Solve $J_i^{(n)} \delta u_i = -F_i^{(n)}$ by GMRES(m)
      Compute $C_i^{(n)}(J_i^{(n)}, F_i^{(n)})$
      If opt=0 **then**
        $G_i^{(n)} = C_i^{(n)}$
      else
        $G_i^{(n)} = G_{i-1}^{(n)} C_i^{(n)}$
      endif
    else

Solve $J_i^{(n)} G_{i^*}^{(n^*)} (G_{i^*}^{(n^*)})^{-1} \delta u_i = -F_i^{(n)}$ by GMRES(m)
endif
$u_{i+1} = u_i + \delta u_i$
$u(n+1) = u_{i+1}$

Here, we test the efficiency of the preconditioner across the time steps using an implicit Euler scheme. In the first test cases, each time step is solved by just one Newton iteration. We build the preconditioner after the first cycle of GMRES(m) and we apply it to solve the following time steps. As can be seen in tables 3 and 4, GMRES(m) is accelerated during the first time steps. In table 5, we update the preconditioner at the fourth time step, improving the efficiency of the preconditioner. In tables 6-9, we consider two Newton iterations, we compute a preconditioner at the first cycle of GMRES for each Newton iteration. In table 7 and 9, we consider the algorithm acc-time(1). In table 6 and 8, we more simply consider the algorithm acc-time(0). We note that for the first test case, Newton doesn't converge thus it is difficult to see what kind of preconditioner for the acceleration is the best.

				Acceleration		
n	$\frac{\|F(u_1)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_0)\|_2}$	J*v	$\frac{\|F(u_1)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_0)\|_2}$	J*v
0	6.312579E-05	5.469117E-05	6	6.312579E-05	5.469298E-05	6
1	1.667603E-02	4.033682E-05	6	1.667898E-02	7.933922E-05	2
2	1.711056E-02	8.067259E-05	6	1.713215E-02	5.281140E-05	5
3	1.563396E-03	5.407076E-05	6	1.567406E-03	7.590680E-05	4
4	1.832167E-02	4.368422E-05	6	1.831945E-02	9.717254E-05	4

Table 3: Observations for problem 1 with a backward Euler scheme when $m = 10$ and $CFL = 1.0$, (one Newton iteration, $freq \geq 5$).

				Acceleration		
n	$\frac{\|F(u_1)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_0)\|_2}$	J*v	$\frac{\|F(u_1)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_0)\|_2}$	J*v
0	1.350409E-02	8.688418E-04	11	1.350409E-02	8.688418E-04	11
1	1.267828E-02	9.853834E-04	14	1.266203E-02	9.089535E-04	11
2	1.106648E-02	7.653879E-04	15	1.104800E-02	7.666862E-04	14
3	1.004400E-02	9.094555E-04	14	9.998064E-03	8.792788E-04	15
4	9.134224E-03	9.246127E-04	14	9.113527E-03	9.050189E-04	16
5	8.804517E-03	8.285131E-04	14	8.810355E-03	8.705162E-04	17

Table 4: Observations for problem 2 with a backward Euler scheme when $m = 15$ and $CFL = 5.0$, (one Newton iteration, $freq \geq 6$).

				Acceleration		
n	$\frac{\|F(u_1)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_0\|_2}$	J*V	$\frac{\|F(u_1)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_0)\|_2}$	J*V
0	1.350409E-02	8.688418E-04	11	1.350409E-02	8.688418E-04	11
1	1.267828E-02	9.853834E-04	14	1.266203E-02	9.089535E-04	11
2	1.106648E-02	7.653879E-04	15	1.104800E-02	7.666862E-04	14
3	1.004400E-02	9.094555E-04	14	1.005455E-02	9.396960E-04	14
4	9.134224E-03	9.246127E-04	14	9.118419E-03	8.903471E-04	9
5	8.804517E-03	8.285131E-04	14	8.735657E-03	8.166246E-04	12

Table 5: Observations for problem 2 with a backward Euler scheme when $m = 15$ and $CFL = 5.0$, (one Newton iteration, $freq = 3$).

					Acceleration		
n	i	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*V	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*V
0	0	1.0	8.688418E-04	11	1.0	8.688418E-04	11
	1	1.350409E-02	7.699990E-04	23	1.350409E-02	9.472127E-04	22
	2	1.048916E-05			1.289427E-05		
1	0	1.0	7.296029E-04	13	1.0	9.830272E-04	12
	1	1.274907E-02	9.412087E-04	22	1.280563E-02	8.201650E-04	15
	2	1.205738E-05			1.066222E-05		
2	0	1.0	7.847132E-04	13	1.0	7.746524E-04	11
	1	1.114887E-02	9.568749E-04	22	1.109384E-02	9.213040E-04	17
	2	1.074007E-05			1.026940E-05		

Table 6: Observations for problem 2 with a backward Euler scheme when $m = 15$ and $CFL = 5.0$.

					Acceleration		
n	i	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*V	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*V
0	0	1.0	8.688418E-04	11	1.0	8.688418E-04	11
	1	1.350409E-02	7.699990E-04	23	1.350409E-02	8.904531E-04	20
	2	1.048916E-05			1.216324E-05		
1	0	1.0	7.296029E-04	13	1.0	9.861310E-04	12
	1	1.274907E-02	9.412087E-04	22	1.280669E-02	9.483351E-04	18
	2	1.205738E-05			1.228417E-05		
2	0	1.0	7.847132E-04	13	1.0	7.983771E-04	11
	1	1.114887E-02	9.568749E-04	22	1.109521E-02	9.150727E-04	19
	2	1.074007E-05			1.024130E-05		

Table 7: Observations for problem 2 with a backward Euler scheme when $m = 15$ and $CFL = 5.0$.

					Acceleration		
n	i	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*v	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*v
0	0	1.0	5.469117E-05	6	1.0	5.469298E-05	6
	1	6.312579E-05	9.655995E-05	11	6.312579E-05	4.688787E-05	12
	2	6.891310E-09			3.978817E-09		
1	0	1.0	4.039129E-05	6	1.0	4.039337E-05	6
	1	1.667599E-02	5.566767E-05	9	1.667599E-02	6.163422E-05	9
	2	1.667382E-02			1.667382E-02		
2	0	1.0	5.292650E-05	6	1.0	5.442649E-05	4
	1	1.513957E-03	4.352684E-05	10	1.512630E-03	7.441549E-05	10
	2	1.512203E-03			1.512202E-03		

Table 8: Observations for problem 1 with a backward Euler scheme when $m = 10$ and $CFL = 1.0$.

					Acceleration		
n	i	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*v	$\frac{\|F(u_i)\|_2}{\|F(u_0)\|_2}$	$\frac{\rho_m}{\|F(u_i)\|_2}$	J*v
0	0	1.0	5.469117E-05	6	1.0	5.469298E-05	6
	1	6.312579E-05	9.655995E-05	11	6.312579E-05	4.071232E-05	12
	2	6.891310E-09			3.744592E-09		
1	0	1.0	4.039129E-05	6	1.0	4.039258E-05	6
	1	1.667599E-02	5.566767E-05	9	1.667599E-02	6.046894E-05	10
	2	1.667382E-02			1.667382E-02		
2	0	1.0	5.292650E-05	6	1.0	5.442932E-05	4
	1	1.513957E-03	4.352684E-05	10	1.512630E-03	8.964561E-05	11
	2	1.512203E-03			1.512204E-03		

Table 9: Observations for problem 1 with a backward Euler scheme when $m = 10$ and $CFL = 1.0$.

4.2 Full computations

We observed from the previous local tests that the preconditioner is efficient across time steps. Now, we want to consider the preconditioner with the same practical approach on full test cases. Furthermore, we will consider test cases when implicit schemes are better than explicit schemes.

4.2.1 Unsteady computation

In this subsection, we compare the explicit methods with the implicit methods when the problem is unsteady. As far as possible, we will compare the methods controlling the local error per unit time step. We consider one unsteady test case which is stiff in the sense that an explicit method is inaccurate.

Error estimator and Stepsize control

To follow the solution, we need accurate schemes, thus we must consider schemes for which the order is greater than one. For explicit methods, we used the well-known fit in Runge-Kutta methods [CM89]. For implicit method, we introduce the BDF schemes, which are unconditionally stable until second order. The class of BDF schemes have the following form

$$u(n+1) = \Delta t_n b_n G(u(n+1)) + \sum_{i=1}^{o} a_{i,n} \Delta t_n u(n-i+1), \tag{14}$$

where o is the order of the scheme. The cost per time step is one non-linear system to be solved forall orders. Once we have computed u an estimation of $u(n+1)$, we estimate the local error err_n defined by $u(t_{n+1}) - \Delta t_n b_n G(u(t_{n+1})) - \sum_{i=1}^{o} a_{i,n} \Delta t_n u(n-i+1)$ where $u(t_{n+1})$ is the exact solution of

$$u_{,t} = G(u), \quad u(t_{n-i+1}) = u(n-i+1), \quad i = 1, \ldots, o$$

by the following procedure.

- Let $o^* = o + 1$ with o given,
- compute $\tau = H(u) = u - \Delta t_n b_n G(u) - \sum_{i=1}^{o} a_{i,n} \Delta t_n u(n-i+1)$,
- compute $H^*(u) = u - \Delta t_n b_n^* G(u) - \sum_{i=1}^{o^*} a_{i,n}^* \Delta t_n u(n-i+1)$,
- $err_n \approx \|H(u) - H^*(u)\|$.

The influence of τ in err_n is measured thanks to this basic result ;

$$\|H(u) - H^*(u) - \{H(u(n+1)) - H^*(u(n+1))\}\|_2 \leq (1 + c_1 c_2)\|\tau\|_2$$

with $c_1 = \|\frac{\partial H^*}{\partial u}(\bar{u})\|_2$, $c_2 = \|\frac{\partial H}{\partial u}^{-1}(\bar{\bar{u}})\|_2$, and $\bar{u}, \bar{\bar{u}}$ belonging to a neighbouring of $u(n+1)$. Once we have estimate the local error, err_n we control the time step Δt_n such that $\frac{err_n}{\Delta t_n} \leq tol$. The control, called EPUS, is attractive as the global error of integration is constant regardless of the number of steps used. Suppose we keep the order of integration constant then the new stepsize is selected by means of the formula

$$\Delta t_{new} = \left(\frac{tol}{err_n}\right)^{\frac{1}{o}} \Delta t_{old},$$

where o is the order of the integration method.

Newton-GMRES parameters control

We choose the Newton tolerance η such that the incomplete Newton convergence does not interfere with the time process. The newton residual at step i and at time t_n is

$$F(u_i) = u_i + \Delta t_n H_1(u_i) - \Delta t_n H_2(u(n), u(n-1), u(n-2), \ldots),$$

with $u_0 = u(n)$, for $i \geq 0$.
Suppose that $||F(u_i)||_2 = \eta ||F(u_0)||_2$ then the mean value theorem gives

$$\eta ||F(u_0)||_2 \leq ||J(\overline{u})||_2 ||u(n+1) - u_i||_2$$

Now, we lay down $||u(n+1) - u_i||_2 \leq 0.01 * tol * \Delta t_n$. So, the tolerance of Newton has to satisfy

$$\eta \leq \frac{0.01 * tol * \Delta t_n * ||J(\overline{u})||_2}{||F(u_0)||_2} \tag{15}$$

The tolerance ξ for GMRES is taken equal to η as we expect a linear behaviour of F in a neighbourhood of $u(n-1)$. To compute $||J(\overline{u})||_2$, we calculate the highest singular value of the Hessenberg matrix $H_k = V_k^t J(u_i) V_k$ estimated in GMRES as the ritz values of J_i, converge to the extremal eigenvalues of $J(u_i)$ as k increases. However, (15) is strong and cannot be satisfied due to parameters constraints (i, m, j_{max}). Thus, we replace (15) by

$$\eta \leq \max(\frac{0.01 * tol * \Delta t_n * ||J(\overline{u})||_2}{||F(u_0)||_2}, \eta_{old})$$

where η_{old} is the last norm residual obtained by Newton-GMRES for the solution of (14) satisfying the local error control.

4.2.2 Steady computation

For explicit schemes, we use the Euler scheme. The choice is motivated by the following consideration. High order explicit schemes are no more efficient as stability is only slightly improved, compared to the increase of the computation. For implicit schemes, we use a second order BDF scheme with a cost of one non linear system to be solved. Numerical results show that using this scheme improve the convergence of Newton algorithm compared to Backward Euler. Controlling the local time step is difficult here as the influence of the error during time integration on the convergence to the steady state is not well-known. So, we will use an empirical law which is test dependent.

4.2.3 Tests

Figure reading for the diagrams

All the computations are done on one processor of the CRAY C98. In order to analyse the evolution of the integration procedure, we have displayed some helpful parameters as

o_e the order for the Runge-Kutta method,
o_i the order for the BDF method,
η the tolerance for Newton,
ξ the tolerance for GMRES,
k the size of the Krylov subspace used for the preconditioner,
m the size of the Krylov subspace,
j_{max} the maximum number of restarting of GMRES(m),
CFL_{max} the maximum CFL number,
tol the tolerance for the error per unit step (EPUS).

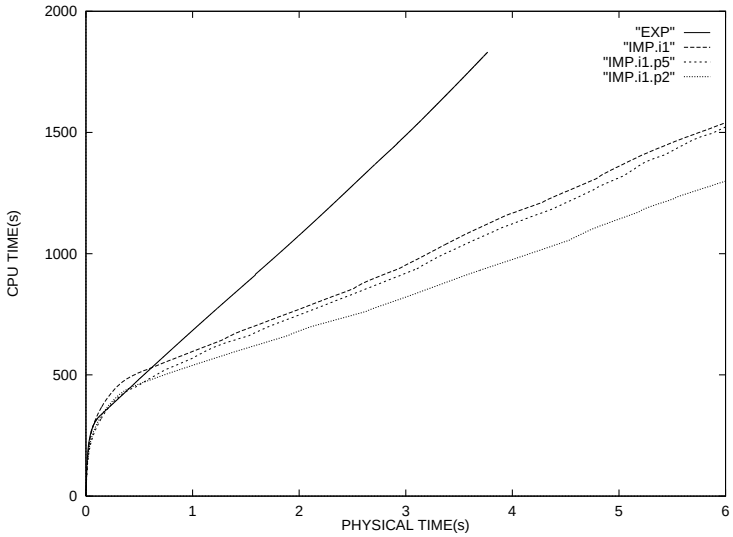
The diagrams in Figures 1 and 3 are composed of curves. The name of each curve (located at the right upper corner) indicates the computation according to the following legend ; **EXP** indicates an explicit computation, **IMP(.i*a*)(.p*b*)** indicates an implicit computation where a is the maximum number of newton steps and b is the frequency for the computation of the preconditioner.

Subsonic unsteady flow over a Naca 0012

We consider a viscous flow $Re = 1000$ over a Naca with a Mach equal to 0.1 and an incidence of 30 degrees. We use a Roe solver. So, the number of unknowns is four times the number of nodes of the mesh. This test case is unsteady and we start to compute the solution from the time $T = 0s$ until $T = 6s$. We consider a mesh compose of 2719 nodes and 5280 triangles. We compare explicit methods with implicit methods using one Newton iteration. We control the local error per unit time step. For explicit methods, we use the well-known fit in Runge-Kutta methods [CM89]. For implicit methods, we use the BDF schemes. We observe an acceleration of the preconditioner clearly shown in the Fig 1.

Subcritical steady flow over a Naca 0012

We have considered the practical value of this approach across a subcritical unviscous flow test around a Naca0012 (801 nodes and 1516 triangles), at free-stream Mach number of 0.63 and 2^o angle of attack (see Figures 4). At a given time step, we use no more than three Newton iterations, we built a preconditioner at each Newton iteration and we used it for the next time steps for the corresponding Newton iteration. To approximate the Euler flux, we used the Osher scheme. For the CFL number, we used the empirical law $CFL = \min(CFL * 2, CFL_{max})$ each 6 iterates for the implicit integration whereas CFL was limited by stability to 0.5 for the explicit integration. An acceleration of the process was observed in Figure 3. Here, the explicit and the implicit computations have a similar cost. As the considered mesh is very coarse, the constraint for the stability of the explicit method is weak thus we expect for a finer mesh that implicit methods will perform better.



o_e	3
o_i	3
k	15
m	15
j_{max}	1
tol	0.05

Figure 1: Comparison of the methods

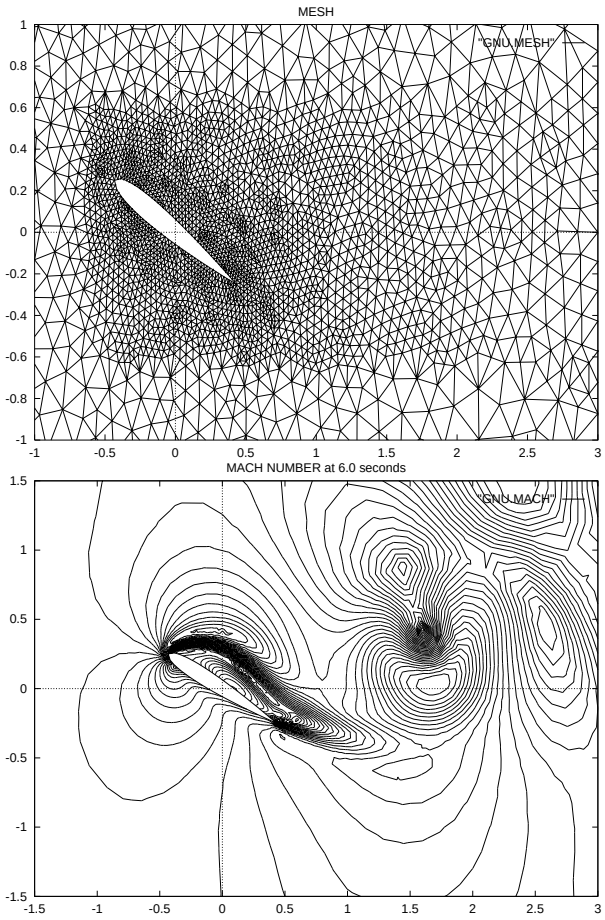
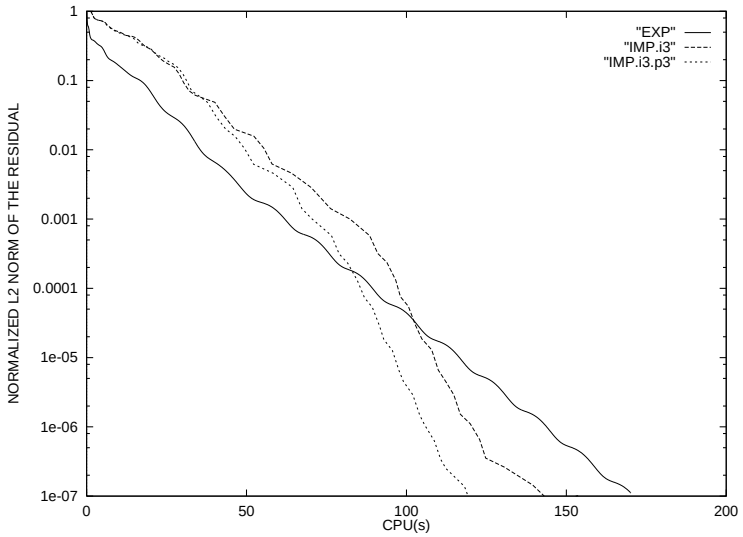


Figure 2: Mesh and Mach number at T=6.s



o_e	1
o_i	2
η	0.01
ξ	0.01
k	15
m	15
j_{max}	2
CFL_{max}	50

Figure 3: Comparison of the methods

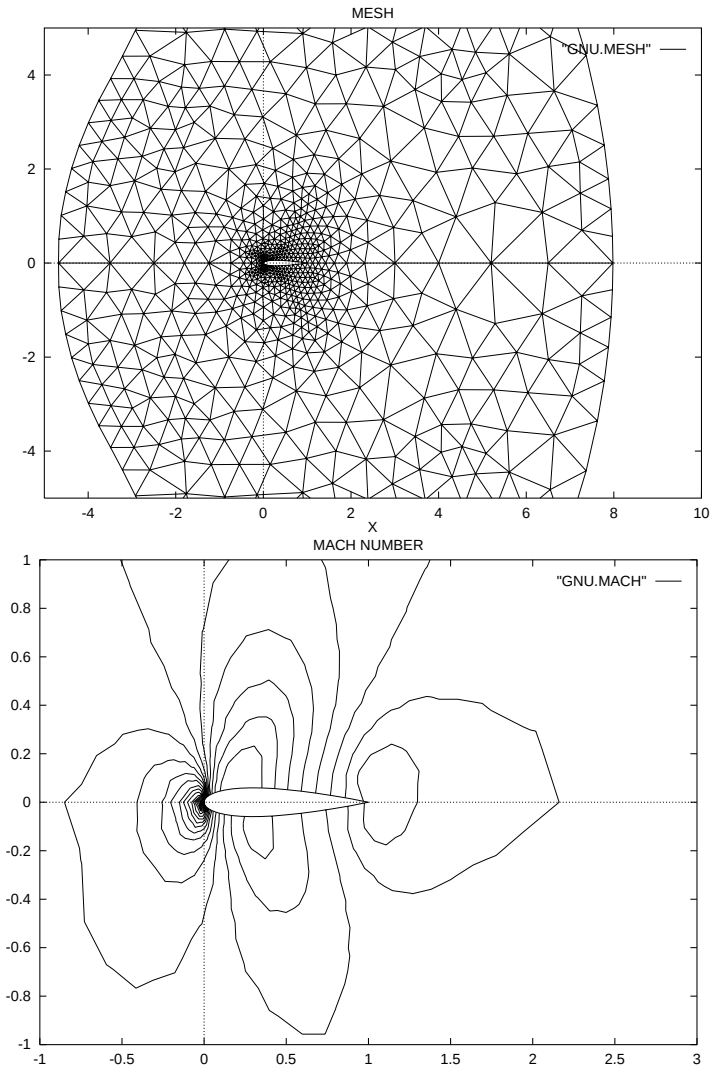


Figure 4: Mesh and Mach number

5 Conclusion

In this paper, we have built a new cheap preconditioner from the GMRES algorithm with the aim to reuse it across the Newton iterations and the time steps. Experimental results show the efficiency of this preconditioner across the time steps and only a slight improvement across the Newton iterations. So, it seems that numerical computations confirm the theorical

result which show that a small variation of the residual is necessary for the preconditioner to be efficient. For unstructured meshes and for unsteady problems, we introduce new implicit schemes for CFD with a control of the error. We have shown within this framework that fast implicit computations are feasible in the stiff part of the integration as the residual varies slowly. New tools have to be developped to switch automatically from explicit method to implicit method. For steady problems, we have shown that the acceleration procedure is always valid although no framework has been defined to characterize the usefulness of implicit methods.

Acknowledgements The author wishes to thank J.Erhel for constructive discussions and necessary guidance for its application.

References

[BS90] P.N Brown and Y Saad. Hybrid krylov methods for nonlinear systems of equations. *SIAM J.Sci.Stat.Comput*, 11(3):450–481, Mai 1990.

[CE93] R Choquet and J Erhel. Some convergence results for the newton-gmres algorithm. Research Report No 755, IRISA, Septembre 1993.

[Cho94] R. Choquet. Acceleration de l'algorithme de newton-gmres pour les equations de navier-stokes. Research Report No 831, IRISA, Juin 1994.

[CM89] M. Crouzeix and A.L. Mignot. *Analyse numerique des equations differentielles, 2nd edition.* Masson, 1989.

[DS83] J.E Dennis and R.B Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall series in Computational Mathematics, 1983.

[Dut91] L.C Dutto. The effect of ordering on preconditioned gmres algorithm applied to solve the compressible navier-stokes equations on unstructured grids. July 1991. Submitted to Int. J. for Numerical Methods in Engineering.

[EBP95] J. Erhel, K. Burrage, and B. Pohl. Restarted gmres preconditioned by deflation. *Journal of computational and applied mathematics*, 1995. To appear.

[HFM86] T.J.R Hughes, L.P Franca, and M Mallet. A new finite element formulation for computational fluid dynamics, i. *Computer Methods in A.M and E*, 54:223–234, 1986.

[HTS93] R. Hixon, F.L Tsung, and L.N Sankar. A comparison of two methods for solving 3-d unsteady compressible viscous flows. In AIAA, editor, *31 Aerospace sciences Meeting, RENO*, 93.

[HW91] E. Hairer and G. Wanner. *Solving ordinary differential equations II. Stiff and differential-algebraic problems. SCM*, Springer-Verlag, 1991.

[JHF91] Z. Johan, T. Hughes, and Shakib F. A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids. *Comp. Meth. in Apllied Mech. and. Eng.*, 87:281–304, 1991.

[Lan94] S. Lanteri. Application du calcul parallele a la resolution d'ecoulements compressibles par une methode d'elements finis. *Calculateurs Paralleles*, 6(4):73–78, 1994.

[LF94] M. Loriot and L. Fezoui. Calcul parallele d'ecoulements compressibles tridimensionels. *Calculateurs Paralleles*, 6(4):85–90, 1994.

[LS88] A Lerat and J Sides. Efficient solution of the steady euler equations with a centered implicit method. *Num.Meth.Fluid.Dyn. K.W. Morton and M.J Baines eds*, III:p 65–86, 1988. T.P Onera 1988-128.

[Moh94] B. Mohammadi. *Fluid Dynamics Computation with NSC2KE. An User-Guide, Release 1.0.* Technical Report RT-0164, INRIA, May 1994.

[OC82] S. Osher and S. Charravarthy. Upwind difference schemes for the hyperbolic systems of conservation law's. *Mathematics of Computation*, April 1982.

[Roe81] P.L. Roe. Approximate riemann solvers, parameters vectors and difference schemes. *Journal of Computational Physics*, 43, 1981.

[Saa87] Youcef Saad. On the lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics of computation*, (178):651–662, April 1987.

[Sen90] Anne-Sophie Sens. *Calcul d'ecoulements transsoniques instationnaires par resolution implicite centree des equations d'Euler sans viscosite artificielle.* PhD thesis, Onera, 1990.

[SF94] A. Soulaimani and M. Fortin. Finite element solution of compressible visous flows using conservative variables. *Comput. Methods. Appl. Mesh. Engrg*, (118):319–350, 1994.

[SS90] G. W. Stewart and Ji-guang Sun. *Matrix perturbation theory.* Academic Press, 1990.

[SW81] J.L Steger and R.F. Warning. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. *Journal of Computational Physics*, (40):263–293, 1981.

[VdVV93] H.A. Van der Vorst and C. Vuik. The superlinear convergence behaviour of gmres. *Journal of Computational and Applied Mathematics*, (48):327–341, 1993.

[Vui93] C Vuik. *Fast iterative solvers for the discretized incompressible Navier-Stokes equations.* Technical Report 93-98, Delft University of Technology, 1993.

[WYY85] L.B Wigton, N.J Yu, and D.P Young. Gmres acceleration of computational fluid dynamics codes. In AIAA, editor, *7th Computational fluid dynamics conference, Ohio*, 85.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399